2026 EDITION

The web3 project launch checklist: idea to mainnet.

Tokenomics, smart contracts, community, go-to-market — the battle-tested phases for launching a web3 project properly. Five phases, the mistakes that kill projects, and the one thing that matters more than any of them.

David Dacruz
Independent AI and web engineer · Bitcoin dev — Ericeira, Portugal

WEEKS 1-4

Most projects die here — before there's even a token.

The foundation phase isn't glamorous. No token, no community, no hype. But everything downstream depends on getting this right. Skip it and you'll spend months fixing problems that should never have existed.

- ☐ **Define the problem in one sentence.** If you can't explain it to someone's grandmother, it's not clear enough. "Decentralized liquidity aggregation layer" isn't a problem — "help people get the best price when they swap" is.
- ☐ **Map the competitive landscape.** Ten projects doing the same thing and you can't articulate the gap? You don't have a project, you have a fork.
- ☐ **Pick one target user.** Not "everyone in crypto." DeFi power users, institutions needing compliance-friendly yield, and gamers who've never held a wallet are three completely different products.
- ☐ **Choose the chain for the use case, not the narrative.** Transaction cost, speed, tooling maturity, where your users already are. Trending ≠ right.
- ☐ **Assemble the core three: builder, community, business.** Three people can launch a web3 project. Three hundred can't if none are the right three.
- ☐ **Get early legal guidance on token classification.** Utility vs security changes distribution, listings, and which jurisdictions you can operate in. Don't wait until launch — by then, answers get expensive.

WEEKS 4–10

Build the product and the token as one system.

A great product with broken tokenomics fails. Great tokenomics with no product fails faster. You design them together because they have to work as one system — incentives aligned with actual usage.

Architecture first

What's on-chain, what's off-chain. How components talk. User flow from wallet connect to transaction complete. Map the whole system before writing a line of Solidity — architecture decisions are expensive to change after deploy.

Ship the MVP

Minimum viable product, not minimum viable promise. A working thing, even rough, is worth more than a beautiful pitch deck. Users can try a product. They can only believe a roadmap.

TOKENOMICS NON-NEGOTIABLES

- ☐ **Utility answers "why hold beyond speculation?"** If the only answer is "number go up," the model is fragile.
- ☐ **Distribution, vesting, emissions — each modeled under bear conditions.** If tokenomics only work when price rises, they don't work.
- ☐ **External review by someone who's watched token economies play out.** Not your co-founder. Not your mod. Fresh eyes catch the assumptions you've stopped questioning.
- ☐ **Smart contract tests: unit, integration, fuzz, edge.** Zero-token sends, drained pools, max supply, reentrancy paths. Bugs found now cost nothing. Bugs found post-deploy cost everything.
- ☐ **Professional security audit budgeted from day one.** $10K–$100K+ depending on complexity. Every major DeFi exploit happened to unaudited or shallowly audited contracts.

WEEKS 8-16

Communities form around a product — not around a token.

Build community while you build the product, not after. The best web3 communities form around something being built, not around a TGE date. Then the weeks before launch are where discipline matters most — pressure is highest, temptation to cut corners is strongest, and cutting them is most expensive.

Community that survives launch

Structured Discord/Telegram — roles, moderation, channels that reflect the user journey. Educational content over hype. Ambassadors who'd use the product without the token. Testnet release so real users find real bugs before mainnet.

Narrative over hype

Why this matters beyond making money. What's broken. How you fix it. People invest in stories. Make yours specific and honest — "number go up" isn't a thesis, it's the absence of one.

PRE-LAUNCH GATE (WEEKS 12-16)

- [ ] **Re-audit if contracts changed.** Then launch a bug bounty. Immunefi or similar. Payout is always cheaper than exploit.
- [ ] **Liquidity strategy written down.** DEX, CEX, or both. How much, from where, at what price range. Thin liquidity = volatile opening = scared holders selling.
- [ ] **Final legal compliance check.** Jurisdictions, disclaimers, token structure. Regulatory problems don't go away — they compound.
- [ ] **Launch-day plan rehearsed.** Who monitors contracts. Who answers the community. Who coordinates exchanges. Who owns the social timeline. Written down. Rehearsed.
- [ ] **Contingency plans for the four common failures.** Contract bug, thin liquidity, exchange delay, coordinated FUD. All of these have happened. Projects with plans survived. Projects with panic didn't.

WEEK 16+

Launch day is the starting line, not the finish.

Everything before was preparation. Now you have real users, real money, real consequences. The first 48 hours need constant vigilance — catch problems in minutes, not hours. And then the second chapter begins, which is where projects either compound or fade.

First 48 hours

TGE executed to plan. Liquidity deployed and monitored for depth/spreads. Contract interactions watched in real time. Alerts wired. Community communication open — the projects that survive rough patches are the ones that talk; the silent ones lose trust fast.

Post-launch discipline

Roadmap execution builds credibility. Every shipped promise compounds. Every missed deadline erodes. Governance introduced when you're ready to do it properly, not as a checkbox. The 90 days after mint matter as much as the six months before it.

MISTAKES THAT KILL PROJECTS

- ☐ **Launching with no product — just a whitepaper and a dream.** Worked in 2021. Does not work in 2026. Ship something real or don't ship.
- ☐ **Over-promised tokenomics that need infinite growth.** Emission curves that look good in a spreadsheet and collapse under real conditions.
- ☐ **"We'll audit after launch."** You'll get hacked before the audit. Millions lost because teams decided one more week was fine.
- ☐ **Treating community as a marketing audience.** Giveaways, hype posts, empty AMAs. They show up for rewards and leave when rewards stop. Build with them, not at them.
- ☐ **No post-launch plan.** All energy goes into launch day. Team exhausted. Roadmap vague. "When next feature?" has no answer. Plan the first 90 days with the same rigor as the launch itself.

Build something that works before you launch a token.

The token serves the product. Not the other way around. If your project only makes sense because of the token, it doesn't make sense at all. The projects that survive cycles are the ones where — if you removed the token entirely — there'd still be something worth using.

- ▢ **The token is a tool.** Incentive design, governance, value capture. Not the product itself.
- ▢ **Product-first teams are still here.** Token-first teams mostly aren't. The pattern is consistent across both bull and bear cycles.
- ▢ **Order matters more than speed.** Foundation → product → tokenomics → community → pre-launch → launch. Skipping steps under pressure is what kills projects, not going slow.
- ▢ **Discipline beats discipline of one.** One founder's rigour won't save a team that cuts corners everywhere else. The whole team has to agree on what "ready" means.
- ▢ **If you're building in web3, start with the product.** Everything else follows from that — and nothing substitutes for it.

LAUNCHING A WEB3 PROJECT? NEED A GUT-CHECK ON THE PLAN?

I've been in web3 since 2021 — through the cycles, through multiple launches. I advise selectively on tokenomics, architecture, launch sequencing, and the product decisions that need to hold after the announcement thread fades.

CONTACT	daviddacruz.dev/contact
WHATSAPP	+33 6 95 43 09 24
MORE	daviddacruz.dev/services/web3-consulting